Utilização da biblioteca *recommenderlab* do Software R para análise comparativa de algoritmos de recomendação

Janderson Jason Barbosa Aguiar[1]

[1] Universidade Federal de Campina Grande (UFCG)
Campina Grande – PB – Brasil

janderson@copin.ufcg.edu.br

Abstract. *Researches on Recommender Systems (RS) are often carried out due to the increasing amount of data that users encounter when accessing the Internet. The R software facilitates the generation of recommendations with the recommenderlab library. In the RS field, it is important to find effective and fast algorithms in various contexts. Therefore, this paper aims to disseminate the use of this library as a means to execute and compare some recommendation algorithms in two different contexts. The recommenderlab library proved to be a facilitator for this comparative analysis, thus it is interesting disclose their use, and the method used in this research, as learning support in the RS field.*

Resumo. *Estudos sobre Sistemas de Recomendação (SR) são frequentemente realizados devido à quantidade crescente de dados que os usuários encontram ao acessarem a Internet. O software R facilita o ato de gerar recomendações a partir da biblioteca recommenderlab. Considerando que, na área de SR, é importante encontrar algoritmos eficazes e velozes em variados contextos, este artigo visa divulgar o uso dessa biblioteca como meio para executar e comparar alguns algoritmos de recomendação em dois contextos diferentes. A biblioteca recommenderlab mostrou-se como facilitadora para a análise comparativa realizada, sendo interessante divulgar seu uso, e o método empregado nesta pesquisa, como meio de apoio ao aprendizado em SR.*

1. Introdução

Devido à grande quantidade de informações e opções existentes na Web, os usuários sentem dificuldade para escolher conteúdos (serviços, produtos, recursos etc.). Como exemplo, o crescente uso de tecnologias de informação e comunicação possibilita mudanças a cada dia no processo de ensino e aprendizagem, sendo desafiador para os professores selecionar e organizar os diversos recursos educacionais que vão surgindo na Web para que os alunos se sintam cada vez mais motivados a aprender [Costa, Aguiar e Magalhães 2013; Aguiar *et al.* 2014]. Para resolver problemas desse tipo, surgiram os Sistemas de Recomendação (SR), despontados na década de 1990 [Cazella, Nunes e Reategui 2010; Bobadilla *et al.* 2013].

O estado da arte em SR mostra que há várias técnicas para recomendar conteúdos [Cazella, Nunes e Reategui 2010; Bobadilla *et al.* 2013], sendo a Filtragem Colaborativa (FC) provavelmente a mais familiar, mais amplamente implementada e mais madura das tecnologias [Burke 2002].

O Ambiente R, ou simplesmente R [R Core Team 2013], é uma importante ferramenta na análise e na manipulação de dados. Consiste em um software de domínio

público que se destaca por seus pacotes estatísticos, mas que também possui outras bibliotecas, com propósitos diversos. Uma dessas bibliotecas é a *recommenderlab* [Hahsler 2011], voltada para desenvolver e testar algoritmos de recomendação, incluindo também alguns algoritmos implementados, tais como FC baseada em Item e FC baseada no Usuário.

Como afirma Hashler (2011), a *recommenderlab* não é uma biblioteca para a criação de aplicações de recomendação, mas fornece uma infraestrutura para realizar pesquisas gerais sobre SR, focando na manipulação consistente e eficiente de dados.

Com base na importância de técnicas de FC e do uso difundido do R, julga-se interessante comparar essas implementações de FC em contextos diferentes para escolher um possível melhor algoritmo que possa ser adotado em pesquisas sobre SR, com a facilidade propiciada pelo software R, nos mais variados contextos — ou seja, sendo útil independente da base de dados utilizada pelo pesquisador.

Tendo em vista essa motivação, foram consideradas estas duas questões de pesquisa:

1. Que algoritmo, dentre os algoritmos de recomendação a serem comparados (disponíveis na *recommenderlab*), apresenta maior eficácia?
2. Qual deles tem maior desempenho (velocidade de recomendação)?

Em linhas gerais, o objetivo da investigação empírica experimental deste trabalho consistiu em analisar algoritmos de recomendação disponíveis em uma biblioteca para uso no software R, com a intenção de compará-los a respeito de sua eficácia e desempenho, do ponto de vista de usuários que recebem recomendações de conteúdo, no contexto dos dados de um site que recomenda piadas (*Jester*) e outro que recomenda filmes (*MovieLens*).

Como resultado geral, os algoritmos se comportaram de maneira diferente dependendo do contexto considerado. Em comparação com recomendação não personalizada (baseada em itens populares) e FC baseada no Usuário (UBCF), os algoritmos de FC baseada no Item (IBCF) apresentaram os piores resultados em relação à eficácia. Em relação à velocidade de recomendação, o resultado variou bastante em relação ao contexto. A *recommenderlab* revelou-se uma biblioteca de fácil uso e, aliada à metodologia empregada para responder as questões de pesquisa citadas acima, defende-se que ela é bastante útil para mediar o processo de ensino–aprendizagem na área de SR.

O artigo está dividido em 5 seções. Na seção 2, são apresentados alguns conceitos sobre SR e trabalhos relacionados. Na seção 3, é tratada a metodologia de comparação dos algoritmos com a biblioteca *recommenderlab*. Na seção 4, são apresentados e discutidos os resultados obtidos a partir da comparação realizada. Na seção 5 encontram-se conclusões e sugestões de trabalhos futuros.

2. Conceitos e Trabalhos Relacionados

Há vários algoritmos relativos à SR e, portanto, é desafiador identificar claramente o melhor algoritmo para um determinado propósito. Uma das razões de ser desafiador é que diferentes algoritmos podem ser melhores ou piores em conjuntos de dados diferentes [Herlocker *et al.* 2004].

Segundo Cazella, Nunes e Reategui (2010), a técnica de FC (considerando a abordagem baseada no algoritmo KNN), uma das mais populares na área de SR, baseia-se essencialmente nestes passos: (i) cálculo do peso de cada usuário em relação à similaridade ao usuário alvo; (ii) seleção de um subconjunto de usuários mais similares (vizinhos) para considerar na predição; e (iii) normalização das avaliações e cálculo das predições ponderando as avaliações dos vizinhos com seus pesos.

Huang (2008) expõe várias medidas de similaridade existentes, tais como: Distância Euclidiana (*Euclidean*), Método dos Cossenos (*Cosine*) e Coeficiente de Jaccard. A Distância Euclidiana consiste na distância entre dois pontos. O Método dos Cossenos consiste em representar os usuários por vetores que contêm as avaliações atribuídas aos itens e a similaridade é dada pelo cosseno do ângulo desses vetores. O Coeficiente de Jaccard consiste no tamanho da interseção dividida pelo tamanho da união dos vetores que representam os usuários.

Wang, De Vries e Reinders (2006) afirmam que as abordagens de FC são geralmente classificadas como baseadas em memória ou baseadas em modelo. Nas abordagens baseadas em memória, as avaliações são armazenadas na memória e, na fase de previsão, os usuários ou itens semelhantes são classificados com base nas avaliações memorizadas; com base nas avaliações desses usuários ou itens semelhantes, são geradas as recomendações para o usuário alvo [Wang, De Vries e Reinders 2006)]. Há várias pesquisas envolvendo a abordagem de FC baseada no usuário, tais como Breese, Heckerman e Kadie (1998), Herlocker *et al.* (1999), Jin, Chai, e Si (2004), Casagrande, Kozima e Willrich (2013), Müller e Silveira (2013), Frade et al (2014).

Sarwar *et al.* (2001) defendem que, em sistemas tradicionais de FC (baseada no usuário), o esforço para realizar as recomendações aumenta com o número de participantes no sistema. Eles realizaram experimentos que sugerem que os algoritmos baseados em itens têm desempenho bem maior do que os algoritmos baseados no usuário, além de proporcionar melhor qualidade. Na mesma linha do trabalho de Sarwar *et al.* (2001), há o trabalho de Deshpande e Karypis (2004), que utiliza uma técnica com base na probabilidade condicional entre itens.

De maneira diferente desses artigos, este trabalho consistiu em comparar as duas abordagens de FC baseada em memória (FC baseada em usuários e FC baseada em itens), variando também a medida de similaridade empregada (*Cosine*, *Euclidean* e Jaccard), e enfocando na implementação de uma biblioteca para um software específico, mas bastante utilizado, o Ambiente R. Além disso, esses algoritmos foram comparados com um algoritmo de recomendação não personalizado, baseado apenas nos itens mais populares (‘POPULAR’) [Hahsler 2011].

3. Metodologia

Com o objetivo de utilizar a biblioteca *recommenderlab* [Hahsler 2011] a partir do software R [R Core Team 2013] para comparar algoritmos de recomendação, foram usados os algoritmos e bases de dados disponíveis nessa biblioteca. Os algoritmos foram de FC baseada no usuário (UBCF) e baseada no item (IBCF), variando-os a partir do método de similaridade empregado (*Cosine*, *Euclidean*, Jaccard). Além desses, foi considerado no experimento o algoritmo de recomendação baseada na popularidade de itens, que gera recomendações considerando unicamente o número de usuários que têm o item em seu perfil. Tais algoritmos são bastante utilizados na área de SR.

Portanto, os fatores (variáveis independentes) do experimento foram:

1. Algoritmo: algoritmo de recomendação que tem como entrada os dados das bases *Jester* ou *MovieLens*. Níveis para este fator: "IBCF *Cosine*", "IBCF *Euclidean*", "IBCF Jaccard", "Popular Items", "UBCF *Cosine*", "UBCF *Euclidean*" e "UBCF Jaccard";
2. Base de dados: conjunto de dados utilizado como entrada para os algoritmos, cujos níveis (*Jester* e *MovieLens*) estão descritos na subseção 3.1;
3. Porcentagem dos dados utilizada para a base de treino: percentual dos dados para o treinamento do algoritmo de recomendação. Os dados restantes consistem na base de teste. Níveis para este fator: 65%, 70%, 75%, 80%.

A partir disso, as variáveis dependentes são métricas relacionadas à eficácia e ao desempenho (valores numéricos, decimais, de natureza quantitativa), descritas na subseção 3.2. Com isso, foi possível responder às questões de pesquisa definidas na seção 1.

Dados os fatores e níveis anteriormente detalhados, tem-se 56 tratamentos, com 30 replicações cada, totalizando 1680 ensaios. O design do experimento utilizado foi Fatorial Completo [Jain 1991], uma vez que o custo para cada execução não foi demasiadamente grande. Em cada replicação do experimento, houve randomização dos conjuntos de treinamento. Outras randomizações não foram necessárias, pois os ensaios eram independentes.

3.1. Conjuntos de Dados

As bases de dados utilizadas foram *Jester Dataset* e *MovieLens Dataset*, abrangendo dois contextos diferentes.

A amostra *Jester Dataset* contempla 5 mil usuários e 100 piadas do site *Jester: The Online Joke Recommender*, coletados entre abril de 1999 e maio de 2003. Todos os usuários selecionados avaliaram pelo menos 36 piadas, com notas entre -10 e 10 (total de 362106 avaliações).

O conjunto *MovieLens Dataset* possui 943 usuários e 1664 filmes do site *MovieLens*, coletados entre setembro de 1997 e abril de 1998. As avaliações foram feitas com notas entre 1 e 5 (total de 99392 avaliações).

3.2. Métricas

As métricas utilizadas para comparar os algoritmos em relação à eficácia e ao desempenho foram as seguintes:

1. *Precision*: quantidade de itens recomendados, que realmente são interessantes para o usuário, em relação ao conjunto de todos os itens que lhe são recomendados;
2. *Recall*: quantidade de itens recomendados, que realmente são interessantes para o usuário, em relação ao conjunto de todos os itens relevantes que poderiam ser recomendados;

3. *F-measure*: média harmônica entre *Precision* e *Recall*;
4. Tempo de resposta: tempo de uma execução do algoritmo de recomendação.

É importante destacar que, em relação ao desempenho, o Tempo de resposta corresponde à soma dos tempos de treino e de predição (em segundos). Em relação à eficácia, os valores de Precisão (*Precision*) e Cobertura (*Recall*) foram calculados a partir das 10 recomendações mais significativas realizadas, e o valor de *F-measure*, também chamado de F_1 *score*, foi calculado a partir da equação (1).

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \quad (1)$$

3.3. Validação

A execução do experimento envolveu os seguintes passos:

1. Coleta dos dados;
2. Separação aleatória dos dados em conjuntos de treinamento e teste;
3. Execução de algoritmos de recomendação para obtenção das métricas;
4. Análise dos resultados.

A análise dos resultados foi estatística, realizando testes com nível de confiança de 95%. Com a definição das questões de pesquisa (na seção 1) e das variáveis independentes e dependentes, foram elaboradas as seguintes hipóteses:

- Hipótese 1: Os valores de *Precision* para os algoritmos de recomendação em estudo são diferentes.
- Hipótese 2: Os valores de *Recall* para os algoritmos de recomendação em estudo são diferentes.
- Hipótese 3: Os valores de *F-measure* para os algoritmos de recomendação em estudo são diferentes.
- Hipótese 4: Os valores de Tempo de resposta para os algoritmos de recomendação em estudo são diferentes.

Para testar estatisticamente as hipóteses definidas, visando comparar os algoritmos, foi utilizada a técnica de análise de variância (*Analysis of Variance* — ANOVA) [Jain 1991]. O teste ANOVA tem como pré-requisito uma análise residual, portanto, para cada uma das variáveis dependentes, verificou-se (i) normalidade dos resíduos, (ii) independência dos erros dos resíduos, e (iii) variação constante dos resíduos.

Para as variáveis relacionadas à eficácia (*Precision*, *Recall* e *F-measure*), as premissas foram satisfeitas, havendo apenas poucos desvios de normalidade. Apesar da ANOVA ser resistente a afastamentos da distribuição normal, a partir das análises residuais, as premissas foram violadas para a variável Tempo de resposta. Portanto, para esta variável, foi realizado o teste de Kruskal-Wallis (teste não paramétrico equivalente à ANOVA) [Boslaugh e Watters 2008]. Todavia, os resultados de ambos os testes indicaram os mesmos resultados.

A partir da verificação das hipóteses, foram gerados intervalos de confiança para as médias, com nível de significância de 5%, além de testes T (paramétrico) e Mann-Whitney U (não paramétrico) [Boslaugh e Watters 2008], par a par (*pairwise*), sendo necessária a identificação da normalidade e da homoscedasticidade dos dados. Foram utilizados os testes de Shapiro-Wilk e de Anderson-Darling [Boslaugh e Watters 2008], para testar normalidade, e os testes de Levene e de Bartlett [Boslaugh e Watters 2008], para homoscedasticidade.

Em síntese, foi possível considerar que, com exceção dos dados referentes a Tempo de resposta, os dados vêm de uma população que tem distribuição normal. Portanto, para *Precision*, *Recall* e *F-measure*, foram aplicados testes T e, para Tempo de resposta, foram aplicados testes Mann-Whitney U. Os resultados encontram-se na seção 4.

4. Resultados e Discussão

No experimento realizado, foram considerados três efeitos principais (α, β, γ), três interações entre dois fatores (αβ, αγ, βγ) e uma interação entre três fatores (αβγ), sendo α o efeito estimado do algoritmo, β o efeito estimado da base de dados, e γ o efeito estimado da porcentagem utilizada para treino. As interações αβ, αγ, βγ e αβγ são relativas aos fatores primários, e o efeito do erro experimental é indicado por ε.

Pela Tabela 1, percebe-se que, para todas as variáveis dependentes, apenas os fatores α e β e a interação αβ contribuem mais fortemente para explicar a variação. É importante destacar que, para todas as variáveis dependentes, o erro experimental é quase desprezível (não atingindo sequer 1%). Os fatores dos efeitos α e β e da interação αβ (algoritmo e base de dados) explicam mais de 99% da variância em todas as variáveis dependentes do experimento. Sendo assim, é possível dizer que a porcentagem utilizada para treino é um fator sem muita importância.

Na Tabela 1 ainda é apresentado, para cada fator e interação entre fatores, o resultado de um teste *F* [Jain 1991]. As hipóteses desse teste consistem em:

- H0: Não existe diferença entre os efeitos das alternativas do fator na variável dependente.
- H1: Existe diferença entre os efeitos das alternativas do fator na variável dependente.

Quando o valor de *F* é maior que o valor de *F-Table*, a hipótese H0 é rejeitada, sendo possível dizer que existe diferença entre os efeitos das alternativas do fator para determinada variável dependente [Jain 1991]. Focando nas linhas da Tabela 1 que mostram os resultados de α, β e αβ, que demonstraram mais importância, é possível notar que o valor de *F* é bem maior que o valor de *F-Table* para todas as variáveis dependentes. Portanto, é possível dizer que os fatores e interação realmente importantes (α, β e αβ) possuem significância estatística, existindo diferença entre os efeitos das alternativas do algoritmo em relação à eficácia (*Precision*, *Recall*, *F-measure*) e ao desempenho (Tempo de resposta). Vale lembrar que, para o Tempo de resposta, como abordado na subseção 3.3, as premissas da ANOVA, para usar o teste *F*, não foram bem satisfeitas, mas, usando o teste de Kruskal-Wallis foi possível chegar à mesma conclusão.

Tabela 1. Alocação de variação e significância dos efeitos do experimento

	Precision			***Recall***		
	SS (%)	*F*	*F-Table*	*SS (%)*	*F*	*F-Table*
α	63,2331	30.043,3	2,1042	12,1878	46.325,1	2,1042
β	26,7297	76.198,9	3,8472	84,1049	1.918.061,7	3,8472
γ	0,0072	6,8	2,6104	0,0007	5,5	2,6104
αβ	9,4490	4.489,4	2,1042	3,6338	13.812,0	2,1042
αγ	0,0030	0,5	1,6102	0,0007	0,9	1,6102
βγ	0,0034	3,3	2,6104	0,0001	1,0	2,6104
αβγ	0,0049	0,8	1,6102	0,0007	0,8	1,6102
ε	0,5697	-	-	0,0712	-	-
	F-measure			**Tempo de resposta**		
	SS (%)	*F*	*F-Table*	*SS (%)*	*F*	*F-Table*
α	49.774,6	2,1042	49.774,6	2,1042	49.774,6	2,1042
β	1.626.125,0	3,8472	1.626.125,0	3,8472	1.626.125,0	3,8472
γ	3,2	2,6104	3,2	2,6104	3,2	2,6104
αβ	11.661,2	2,1042	11.661,2	2,1042	11.661,2	2,1042
αγ	0,9	1,6102	0,9	1,6102	0,9	1,6102
βγ	2,6	2,6104	2,6	2,6104	2,6	2,6104
αβγ	0,9	1,6102	0,9	1,6102	0,9	1,6102
ε	-	-	-	-	-	-

Considerando a existência de diferença entre os efeitos das alternativas do algoritmo em relação à *Precision*, *Recall*, *F-measure* e Tempo de resposta, é possível aceitar todas as hipóteses definidas na subseção 3.3.

Depois disso, os intervalos de confiança (ICs) para as médias, apresentados nas Figuras 1, 2, 3 e 4, e os testes par a par indicaram as diferenças entre os algoritmos.

Vale ressaltar que quanto maiores os valores de *Precision*, *Recall* e *F-measure*, maior a eficácia; e quanto menor os valores para Tempo de resposta, maior a velocidade de recomendação (desempenho). A partir dos intervalos de confiança e dos testes par a par, foi possível definir *rankings* dos algoritmos com melhores resultados em relação à eficácia e ao desempenho (Tabela 2).

No contexto de um site que recomenda piadas (*Jester*), o algoritmo baseado em itens populares é o que apresenta maior eficácia, seguido dos algoritmos UBCF e, por último, os algoritmos IBCF. Em relação à velocidade de recomendação, ainda nesse contexto, o algoritmo IBCF, considerando o método Jaccard de similaridade, é o que apresenta melhor desempenho, seguido dos outros algoritmos IBCF e baseado em itens populares, e, por último, os algoritmos UBCF.

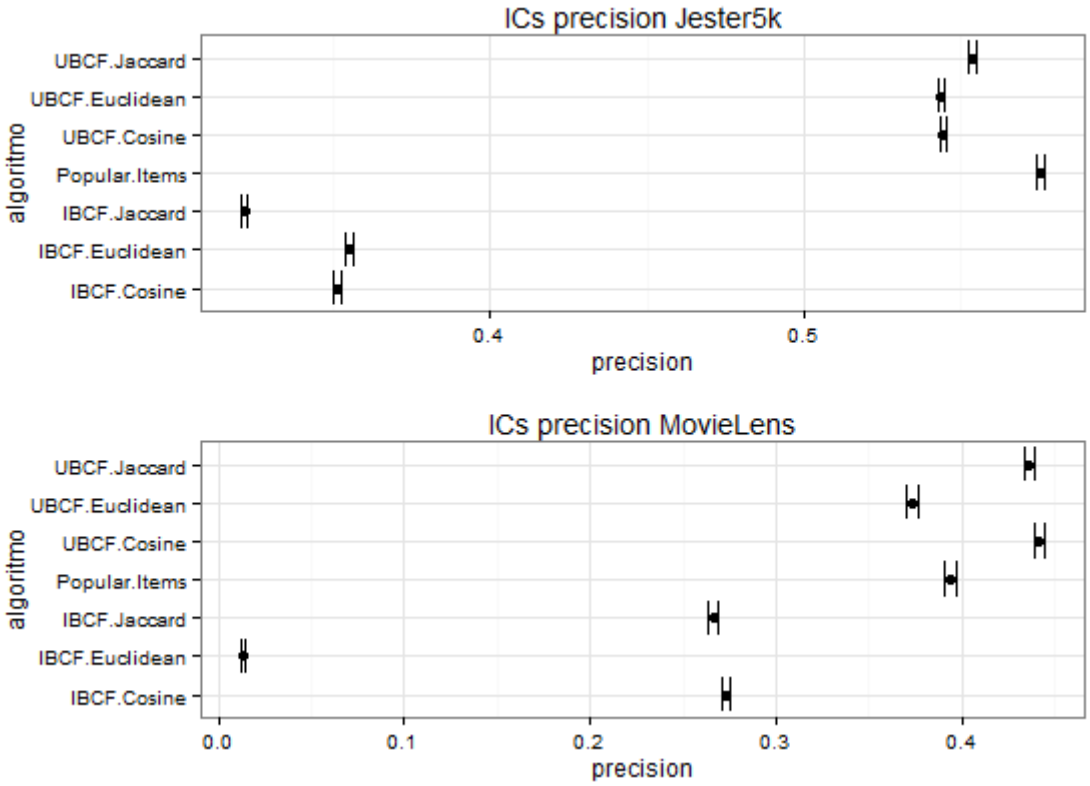

Figura 1. Intervalos de confiança para média relativos a *Precision*

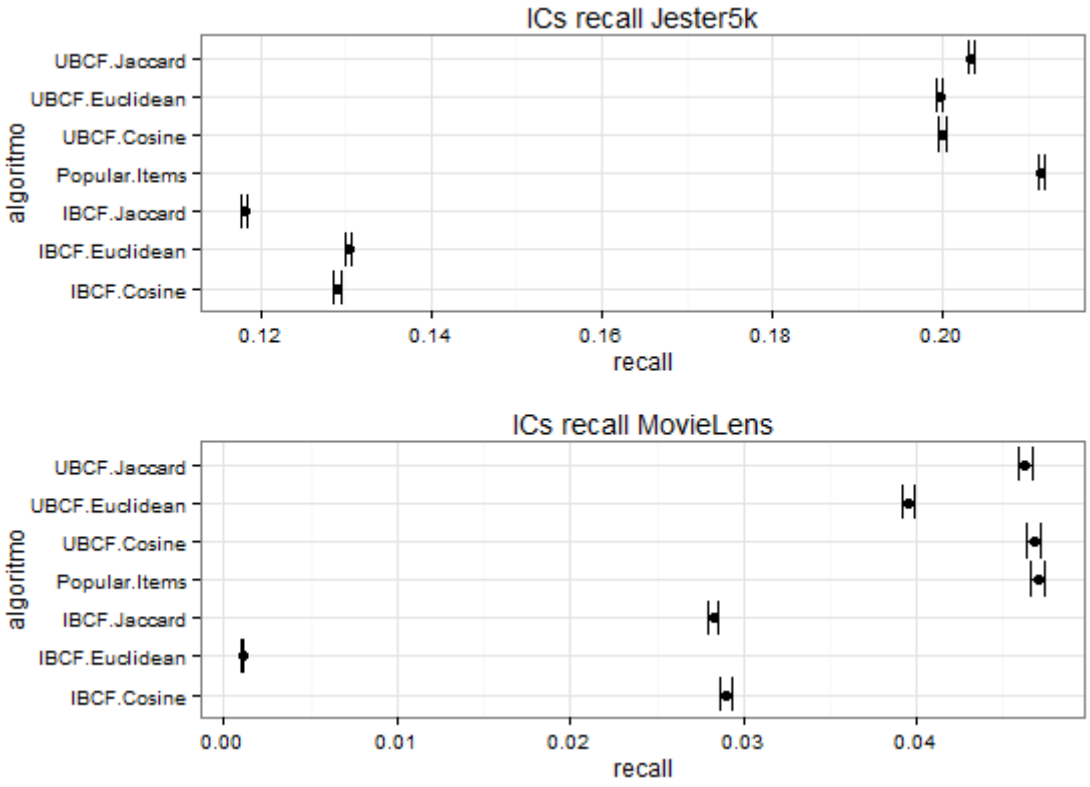

Figura 2. Intervalos de confiança para média relativos a *Recall*

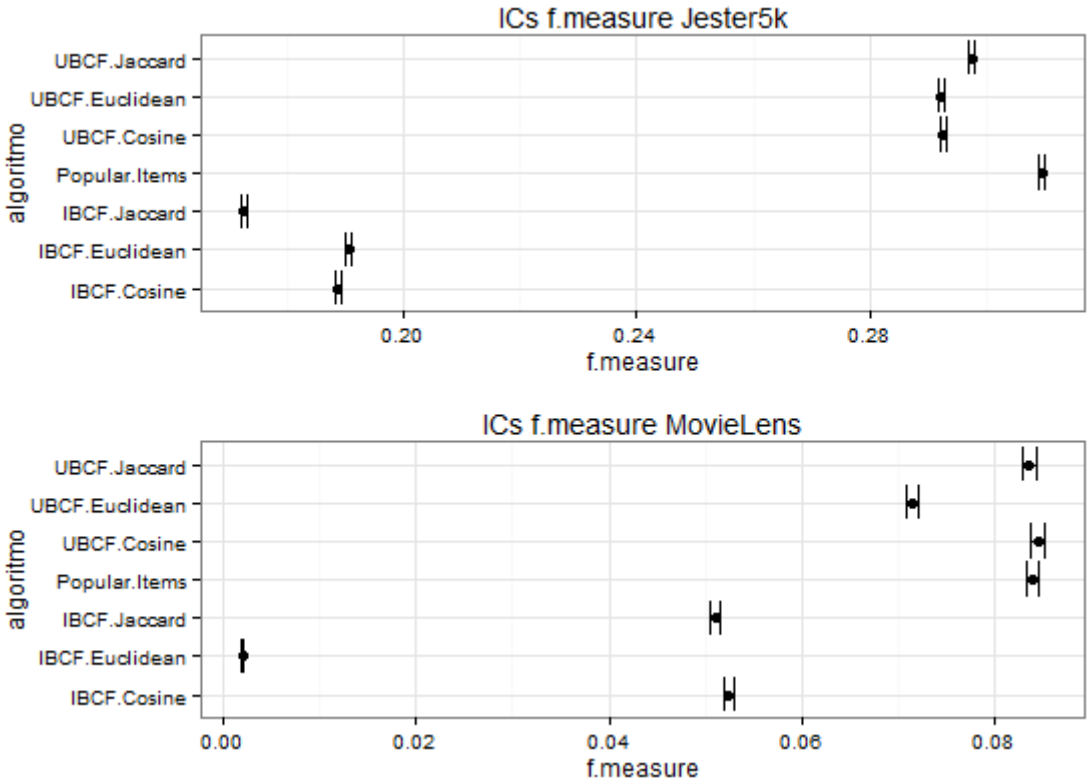

Figura 3. Intervalos de confiança para média relativos a *F-measure*

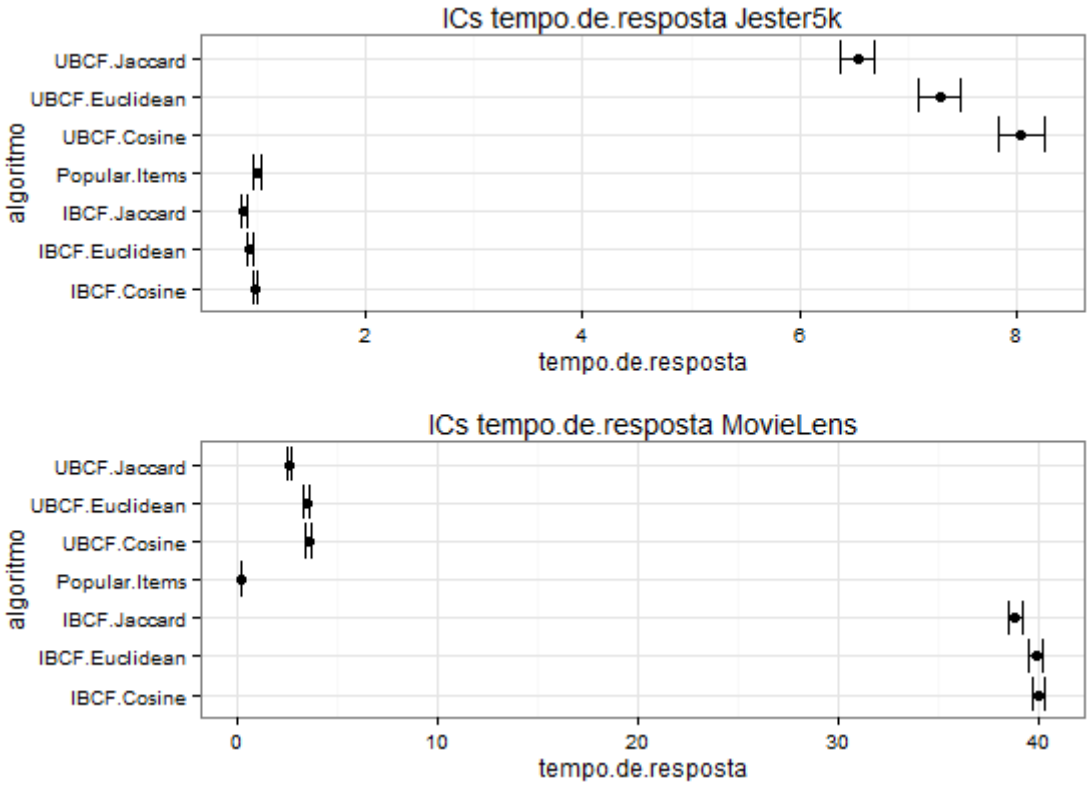

Figura 4. Intervalos de confiança para média relativos a Tempo de resposta

Tabela 2. *Ranking* dos algoritmos em relação à eficácia e ao desempenho

	Precision		***Recall***	
	Jester5k	*MovieLens*	*Jester5k*	*MovieLens*
1º	Popular.Items	UBCF.Cosine	Popular.Items	Popular.Items UBCF.Cosine UBCF.Jaccard
2º	UBCF.Jaccard	UBCF.Jaccard	UBCF.Jaccard	UBCF.Euclidean
3º	UBCF.Cosine UBCF.Euclidean	Popular.Items	UBCF.Cosine UBCF.Euclidean	IBCF.Cosine
4º	IBCF.Euclidean	UBCF.Euclidean	IBCF.Euclidean	IBCF.Jaccard
5º	IBCF.Cosine	IBCF.Cosine	IBCF.Cosine	IBCF.Euclidean
6º	IBCF.Jaccard	IBCF.Jaccard	IBCF.Jaccard	-
7º	-	IBCF.Euclidean	-	-

	F-measure		**Tempo de Resposta**	
	Jester5k	*MovieLens*	*Jester5k*	*MovieLens*
1º	Popular.Items	UBCF.Cosine Popular.Items UBCF.Jaccard	IBCF.Jaccard	Popular.Items
2º	UBCF.Jaccard	UBCF.Euclidean	IBCF.Euclidean IBCF.Cosine Popular.Items	UBCF.Jaccard
3º	UBCF.Cosine UBCF.Euclidean	IBCF.Cosine	UBCF.Jaccard	UBCF.Euclidean UBCF.Cosine
4º	IBCF.Euclidean	IBCF.Jaccard	UBCF.Euclidean	IBCF.Jaccard
5º	IBCF.Cosine	IBCF.Euclidean	UBCF.Cosine	IBCF.Euclidean IBCF.Cosine
6º	IBCF.Jaccard	-	-	-
7º	-	-	-	-

No contexto de um site que recomenda filmes (*MovieLens*), não fica muito nítido definir qual algoritmo se sobressai aos demais, dadas as variações em relação às variáveis *Precision*, *Recall* e *F-measure*. Mas, de forma geral, é possível dizer que o algoritmo baseado em itens populares e os algoritmos UBCF se sobressaem aos algoritmos IBCF em relação à eficácia. Em relação à velocidade de recomendação, ainda nesse contexto, o algoritmo baseado em itens populares é o que apresenta maior desempenho em relação à velocidade de recomendação, seguido dos algoritmos UBCF e baseado em itens populares, e, por último, os algoritmos IBCF.

Além disso, foi observado que os métodos de similaridade empregados nos algoritmos UBCF e IBCF apresentam diferenças entre si, mas variando bastante em relação à métrica e ao contexto. Em relação ao Tempo de resposta, o método de Jaccard sobressai em relação ao método de Distância Euclidiana e ao Método dos Cossenos,

sendo este último, em média, o mais demorado. Em relação à eficácia, não há um método que sempre se sobressai, mas é possível dizer que, em relação à estratégia UBCF, o método *Euclidean* é o que apresenta, em média, piores resultados.

5. Conclusões

Este trabalho consistiu na utilização da biblioteca *recommenderlab* para a realização de um experimento comparativo de algoritmos de recomendação baseada em FC e em itens populares, a respeito de sua eficácia e desempenho, considerando dois contextos diferentes (piadas e filmes). A partir dos resultados, é possível dizer que os algoritmos em estudo se comportaram de maneira diferente em relação à eficácia e à velocidade de recomendação (desempenho). O comportamento também varia dependendo do contexto (base de dados), mas não varia significativamente em relação à porcentagem do conjunto de treinamento (65% a 80%).

Em relação à eficácia, os algoritmos IBCF são os que apresentam piores resultados. Dada a questão de pesquisa "Que algoritmo, dentre os algoritmos de recomendação a serem comparados (disponíveis na *recommenderlab*), apresenta maior eficácia?", é possível dizer que há certa variação em relação ao contexto, mas é interessante destacar o algoritmo baseado em itens populares.

Em relação à velocidade de recomendação, o resultado varia bastante em relação ao contexto (os algoritmos IBCF, por exemplo, estão nas primeiras colocações em um contexto, e nas piores colocações em outro contexto, devido à diferença da quantidade de itens nas bases de dados). Dada a questão de pesquisa "Que algoritmo, dentre os algoritmos de recomendação a serem comparados (disponíveis na recommenderlab), apresenta maior desempenho (velocidade de recomendação)?", é interessante considerar o algoritmo baseado em itens populares como uma boa alternativa em relação ao Tempo de resposta ao variar o contexto.

Além de indicar os algoritmos de recomendação que se destacam positivamente e negativamente em relação às diferentes métricas nos contextos utilizados, outra contribuição do experimento consiste em poder indicar o uso do algoritmo baseado em itens populares, dentre os algoritmos de recomendação comparados, como uma boa opção para ambos os contextos e, possivelmente, para outros tipos de conteúdo.

Sobre as limitações deste trabalho, é possível dizer que, em relação a ameaças à validade externa, a pesquisa não pode ser generalizada além dos contextos em estudo (site que recomenda piadas e site que recomenda filmes). Além disso, em relação a ameaças à validade de constructo, é possível que a baixa significância do fator "porcentagem utilizada para treino" seja por causa dos níveis escolhidos.

Portanto, como sugestão para trabalhos futuros, outro estudo poderia ser realizado com níveis maiores (90%, por exemplo) e menores (50%, por exemplo) para o fator "Porcentagem dos dados utilizada para a base de treino". Além disso, embora a utilização de duas bases de dados distintas seja uma maneira de não restringir muito o escopo, outros estudos poderiam considerar outras bases de dados, viabilizando maior generalização dos resultados obtidos.

Os detalhes em relação à metodologia utilizada (a exemplo da especificação do Design Fatorial Completo e dos testes estatísticos utilizados), além de referências sobre

esses métodos, foram comentados neste artigo objetivando facilitar futuros pesquisadores a adotar tal metodologia também para realizar comparações similares.

Mesmo que a biblioteca *recommenderlab* não seja para criar aplicações de recomendação, como defende seu idealizador [Hashler 2011], seu uso é importante por poder facilitar estudos em nível inicial e intermediário para curiosos da área de SR. Além disso, pode facilitar para os docentes dessa área abordar o conteúdo de SR, fazendo com que os alunos pratiquem/experimentem conceitos importantes, como FC, ao mesmo tempo que utilizam uma ferramenta estatística muito conhecida para análise de dados (o software R).

Espera-se, com este artigo, motivar mais pesquisadores em Tecnologia de Informação e Comunicação para estudar sobre recomendações de itens, especialmente devido à facilidade que a biblioteca *recommenderlab* propicia para iniciantes, e devido à vasta variedade de contextos que se pode alcançar em SR além dos tratados neste artigo (piadas e filmes), tais como materiais didáticos, pessoas etc.

Referências

Aguiar, J. J. B.; Santos, S. I. N.; Fechine, J. M.; Costa, E. (2014). Um Mapeamento Sistemático sobre Iniciativas Brasileiras em Sistemas de Recomendação Educacionais. In: *Anais do 25º Simpósio Brasileiro de Informática na Educação, Dourados*, p. 1123–1132, http://www.br-ie.org/pub/index.php/sbie/article/view/3058/2566.

Bobadilla, J.; Ortega, F.; Hernando, A.; Gutiérrez, A. (2013). Recommender systems survey. In *Knowledge Based Systems*, v. 46, p. 109–132.

Boslaugh, S.; Watters, P. A. (2008). Statistics in a Nutshell. O'Reilly. ISBN: 978-0-596-51049-7.

Breese, J. S.; Heckerman, D.; Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proc. of UAI*, http://research.microsoft.com/en-us/um/people/heckerman/bhk98uai.pdf.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, Dordrecht, v. 12, p. 4, p. 331–370, http://dl.acm.org/citation.cfm?id=586352.

Casagrande, M. F. R.; Kozima, G.; Willrich, R. (2013). Técnica de Recomendação Baseada em Metadados para Repositórios Digitais Voltados ao Ensino. In: *Anais do 24º Simpósio Brasileiro de Informática na Educação, Campinas*, p. 677–686, http://www.br-ie.org/pub/index.php/sbie/article/view/2546/2204.

Cazella, S. C.; Nunes, M. A. S. N.; Reategui, E. B. (2010). A Ciência da Opinião: Estado da arte em Sistemas de Recomendação. In *XXX Congresso da Sociedade Brasileira de Computação — Jornada de Atualização em Informática (JAI*), p.161–216, http://200.17.141.213/~gutanunes/hp/publications/JAI4.pdf.

Costa, E.; Aguiar, J.; Magalhães, J. (2013). Sistemas de Recomendação de Recursos Educacionais: conceitos, técnicas e aplicações. In *II Congresso Brasileiro de Informática na Educação — Jornada de Atualização em Informática na Educação (JAIE)*, p.57–78, http://www.br-ie.org/pub/index.php/pie/article/view/2589/2245.

Deshpande, M.; Karypis, G. (2004). Item-based Top-N Recommendation Algorithms. *ACM Trans. Inf. Syst*. 22 (1), 143–177, http://stuyresearch.googlecode.com/hg/blake/resources/10.1.1.102.4451.pdf.

Frade, R. V. C.; Neto, F. M. M.; Lima, R. W.; Lima, R. M.; Silva, L. C. N.; Souza, R. C. (2014). Um Ambiente Virtual 3D Multiagente com Recomendação Personalizada de Objetos de Aprendizagem. In: *Anais do 25º Simpósio Brasileiro de Informática na Educação, Dourados*, p. 1068–1077, http://www.br-ie.org/pub/index.php/sbie/article/view/3050/2560.

Hahsler, M. (2011). *recommenderlab*: A Framework for Developing and Testing Recommendation Algorithms. http://cran.r-project.org/web/packages/*recommenderlab*/vignettes/*recommenderlab*.pdf.

Herlocker, J. L.; Konstan, J. A.; Terveen, L. G.; Riedl, J. T. (2004). Evaluating collaborative Filtering recommender systems. *ACM Trans.Inf. Syst.*, 22(1):5–53, http://dl.acm.org/citation.cfm?id=963772.

Herlocker, J.; Konstan, J.; Borchers, A.; Riedl, J. (1999). An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of the 1999 Conference on Research and Development in Information Retrieval*, http://www.grouplens.org/papers/pdf/algs.pdf.

Huang, A. (2008). Similarity measures for text document clustering. In *New Zealand Computer Science Research Student Conference*, pp. 49–56.

Jain, R. (1991). The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling. *Wiley Computer Publishing, John Wiley & Sons*, Inc. ISBN: 0-471-50336-3. (Errata: http://www.cs.wustl.edu/~jain/books/ftp/errors_all.pdf)

Jin, R.; Chai, J. Y.; Si, L. (2004). An Automatic Weighting Scheme for Collaborative Filtering. In *Proc. of SIGIR*, http://www.cse.msu.edu/~jchai/Papers/SIGIR04.pdf.

Müller, L.; Silveira, M. S. (2013). Podes me ajudar? Apoiando a formação de pares em sistemas de ajuda em pares através de técnicas de recomendação. In: *Anais do 24º Simpósio Brasileiro de Informática na Educação, Campinas*, p. 868–877, http://www.br-ie.org/pub/index.php/sbie/article/view/2565/2223.

R Core Team. (2013). R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria, http://www.R-project.org.

Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, ACM, New York, NY, USA, pp. 285–295, http://www.ra.ethz.ch/cdstore/www10/papers/pdf/p519.pdf.

Wang, J.; de Vries, A. P. ; Reinders, M. J. T. (2006). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM Press, New York, NY, USA, p. 501–508, http://dl.acm.org/citation.cfm?id=1148257.